UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/675,915	10/01/2003	Youval Bronicki	1577	2086

62433 7590 12/21/2007

EDWARD LANGER
c/o SHIBOLETH YISRAELI ROBERTS ZISMAN & CO.
1 PENN PLAZA-SUITE 2527
NEW YORK, NY 10119

EXAMINER	
WOOD, WILLIAM H	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
12/21/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/675,915	BRONICKI ET AL.	
	Examiner	Art Unit	
	William H. Wood	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 August 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 August 2007* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

WILLIAM WOOD
PRIMARY EXAMINER

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. *20071016*.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claims 1-30 are pending and have been examined.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 21-23 and 30 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 21 and 30 are software per se and therefore not patentable.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1-30 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Independent claims 1 and 30 recited "such that the need for writing code in any programming language to implement the software applications is

eliminated". No support for this can be found in the originally filed disclosure. The original disclosure indicates some form of code writing may very well occur (see at least: page 1, line 8; page 4, line 14; page 4, line 17; page 4, lines 24-27; page 13, lines 5-8). The disclosure make extensive use of "substantially replacing the writing of source code" not completely replacing the writing of source code. Further, the visual modeling Applicant lays claim to is considered simply programming in a particular modeling language and producing code in that language. Therefore, there is no support for not writing code in "any programming language". All the independent claims (and thus the dependent claims) make some use this new and unsupported concept to removing the need to write code. Correction is required.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-12, 14-16 and 30 are rejected under 35 U.S.C. 102(b) as being anticipated by **Williams** (USPN 5,850,548).

Claim 1

Williams disclosed a modeling method for defining software applications using a visualizable computer executable modeling language, said method comprising:

providing a plurality of display elements for displaying screen objects on a display screen *(figure 1, element 106 and figure 2B)*;

displaying components corresponding to said screen objects *(figure 1, element 106 and figure 2B)*;

defining each of the software applications as a hierarchy of process models, slots, data models, and flow rules *(figures 3A-4G; column 2, lines 20-39)*;

classifying some of said process models and said data models as atomic;

classifying all other process models and data models as composite *(figures 3A-4G; column 2, lines 20-39)*;

defining each of said composite process models as a construction of at least one of sub process models, slots, data models, and flow rules *(figures 3A-4G; column 2, lines 20-39)*;

defining a portion of said slots as having one of the following sub-classifications:

mandatory *(figures 3A-B and 18; column 8, lines 45-46; column 17, lines 49-50; thus ports/slots are optional, when not connected and*

mandatory when connected and responding in a mandatory fashion to external events); and

optional *(figures 3A-B and 18; column 8, lines 45-46; column 17, lines 49-50; thus ports/slots are optional, when not connected and mandatory when connected and responding in a mandatory fashion to external events)*;

defining each of said composite data models as a construction of at least one sub data model *(figures 3A-4G; column 2, lines 20-39)*; and

defining each of said flow rules as connecting a pair of said slots, data models and sub data models, wherein said flow rules define both data flow and process flow *(figures 3A-4G; column 2, lines 20-39)*, such that the need for writing code in any programming language to implement the software applications is eliminated *(see above 35 USC 112 rejection; both* ***Williams*** *and the Applicant's original disclosure program in at least a visual language writing at least visual code)*.

Claim 2

Williams disclosed a visualizable computer executable modeling language system operating in accordance with the method of claim 1, for substantially complete definition of the software applications, said system comprising:

process models, each of which may contain any number of sub process models, slots, data models and flow rules *(figures 3A-4G; column 2, lines 20-39)*;

data models, each of which may contain any number of sub data models *(figures 3A-4G; column 2, lines 20-39)*; and

flow rules, each of which connecting a pair o said slots, data models and sub data models, thereby defining data flow and process flow *(figures 3A-4G; column 2, lines 20-39)*;

wherein said process models and data models and slots and flow rules are arranged in a structural hierarchy conforming to a set of rigid composition rules, ensuring the language system is rich enough and precise enough for computer to execute an application model defined in said modeling language *(figures 3A-4G; column 2, lines 20-39)*.

Claim 3

Williams disclosed the modeling language system of claim 2, further comprising at least one visual representation *(figures 3A-4G; column 2, lines 20-39)*.

Claim 4

Williams disclosed the modeling language system of claim 3, wherein said visual representation comprises:

process diagrams comprising various two dimensional shapes representing said process models *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11)*;

sub process diagrams comprising various two dimensional shapes contained within said process diagrams, representing said sub process models *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11)*;

slot diagrams comprising various two dimensional shapes situated on the edges of said process diagrams and said sub process diagrams, representing said slots *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11)*;

data trees comprising hierarchical tree structures contained within said process diagrams, representing said data models and said sub data models *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11)*; and

flow arrows comprising arrows connecting pairs of said slot diagrams, said data trees and sub-tree of said data trees, said arrows representing said flow rules *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11)*.

Claim 5

Williams disclosed the modeling language system of claim 2, wherein each of said slots is further defined as having one of the following classifications:

input slot *(figures 3A and 3B)*; and

output slot (exit) *(figures 3A and 3B)*;

and further defining each of said input slots as having one of the following sub-classifications:

synchronous input slot (trigger) *(figures 3A and 3B; and figure 18, event)*; and

asynchronous input slot *(figures 3A and 3B; and figure 18, event)*;

and further defining each of said triggers as having one of the following sub-classifications:

mandatory *(figures 3A and 3B; and figure 18, event)*; and

optional *(figures 3A and 3B; and figure 18, event)*;

and further defining some of said exits as having the sub-classification terminating *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11)*.

Claim 6

Williams disclosed the modeling language system of claim 2, wherein each of said slots is further defined as having one of the following classifications:

input slot; and

output slot (exit);

and further defining each of said flow rules contained in said composite process models as connecting one source and one target;

and further defining the source of each of said flow rules to be one of the following:

an input slot of said composite process model;

an exit of a sub process model of said composite process model;

a data model of said composite process model; and

a sub data model of a data model of said composite process model;

and further defining the target of each of said flow rules to be one of the following:

an exit of said composite process model;

an input slot of a sub process model of said composite process model;

a data model of said composite process model; and

a sub data model of a data model of said composite process model.

(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11; for all above)

Claim 7

Williams disclosed the modeling language system of claim 2, wherein:

each of said process models may further contain a reference to a database table (process table);

at least some of the sub data models of data models of said process model are marked as interesting fields; and

each of said interesting fields further contains a reference to a column of said process table.

(in addition to figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11; note column 13, line 44 to column 23, line 17; for all above)

Claim 8

Williams disclosed the modeling language system of claim 7, wherein:

a selection condition of an SQL query (addressing clause) may be attached to an input slot of said process model to select matching instances of said process model each time data is to be received by said instances through said input slot;

said addressing clause is defined in terms of a matching condition between the interesting fields of said process model and the data model of said data to be received through said input slot *(column 12, line 60 to column 13, line 10)*.

Claim 9

Williams disclosed the modeling language system of claim 2, wherein each of said process models may further contain a reference to a computer code implementing the function of said process model *(figure 9A)*.

Claim 10

Williams disclosed the modeling language system of claim 2, wherein:

each of said composite data models is composed of said sub data models by one of the following structure means:

concatenation;

collection; and

selection,

each of said sub data models having a classification as one of the following:

mandatory; and

optional;

each of said sub data models may further be marked as recurring, with a further optional indication of minimal and maximal number of occurrences;

each of said data models may further contain constraints on the data it defines, comprising:

at least one of the following:

legal characters; and

minimal and maximal length;

each of said data models may further comprise a set of legal values and an initial value; and

each of said data models may further comprise formatting directives.

(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11; for all above)

Claim 11

Williams disclosed a modeling system for defining software applications using the visualizable computer executable modeling language system of claim 2, wherein enabling users to create, display, modify and test, in an integrated workspace, models of said modeling language, in accordance with the rules of said modeling language system *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 42)*, wherein:

said modeling system comprises a graphical user interface tool (visual modeling tool) for creating, displaying, modifying and testing models of said modeling language in an integrated workspace, such that users of said modeling tool create and edit said models using various graphical user interface (GUI) operations *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 42)*.

Claim 12

Williams disclosed the modeling system of claim 11, wherein each of said process models and said data models is further defined as having one of the following classifications:

dependent model: only exists as sub model of a specific parent model *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11)*; and

reusable model: may be reused as a sub model of multiple parent models, wherein each of said reusable models is assigned a unique identifier *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11).*

Claim 14

Williams disclosed the modeling system of claim 11, further comprising at least the following editing capabilities:

selection of editing operations from menus;

adding components to said models through dragging of models from palettes of existing models; and

modifying attributes of said models and said components of said models *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 42; for all above).*

Claim 15

Williams disclosed the modeling system of claim 11, wherein said workspace comprises a virtually infinite drawing board for displaying hierarchies of two dimensional visual diagrams, each representing a corresponding said hierarchy of models, and wherein said users are able to zoom in an out from a currently displayed part of said hierarchy of diagrams, enabling the display of the details of said model and any sub-model thereof at any desired level of said hierarchy

of models *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 42).*

Claim 16

Williams disclosed the modeling system o claim 11, further comprising:

a software program (runtime engine) to execute models defined in said modeling language; and

a visual debugger for testing and debugging said models, wherein:

said runtime engine, as it executes said models, produces records listing the details of said execution (trace events);

said trace events are used to record and store the history of said execution; and

said visual debugger uses said stored trace events to display the current status of instances of processes, including the content of their data, as well as the processing steps that have led to said current status *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 9, line 11; for all above).*

Claim 30

The limitations of claim 30 correspond to the limitations of claim 1 and are therefore rejected in a corresponding manner.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Williams** (USPN 5,850,548).

Claim 13

Williams disclosed the modeling system of claim 11, wherein models are formally represented as one of the following:

structured database records; and

any other equivalent binary representation,

and wherein a repository of said representations of said models, arranged as a hierarchy of packages and sub-packages (knowledge base), is used to maintain libraries of said models, and wherein the modeling system displays said models whose said representations are stored in said knowledge base, stores in said knowledge base said representations of new said models that are defined by the users of the modeling system, and updates said representations of said models in said knowledge base according to modifications made to said models by said users *(column 2, lines 20-39; and column 5, line 31 to column 13, line 42).*

Williams did not explicitly state XML documents. Official Notice is taken that it was known at the time of invention to utilize XML. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the visual modeling system of **Williams** with XML for storage and documents. This implementation would have been obvious because one of ordinary skill in the art would be motivated to make use of standard technology which is simple and quick to implement (XML).

10. Claims 17-20, 24-25 and 27-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Williams** (USPN 5,850,548) in view of **Parr** et al. (US 2002/0095653).

Claim 17

Williams disclosed each of the applications is defined by a single said process model and a hierarchy of its sub-models *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 42).* **Williams** did not explicitly state the runtime engine executes the application exactly as defined by said single process model and said hierarchy of its sub-models, thus substantially eliminating the need for writing code in any programming language to implement the application. **Parr** demonstrated that it was known at the time of invention to use runtime engines to execute visual programs (page 1, paragraph 0009 to page 2, paragraph 0038). It would have been obvious to

one of ordinary skill in the art at the time of invention to implement the visual model of **Williams** with runtime engine as found in **Parr**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to accessibility to a project as long as possible, up to compilation (**Parr**: page 1, paragraphs 0009-0010).

Claim 18

Williams and **Parr** disclosed the runtime engine of claim 17, further comprising:

active models comprising objects responsible for representing and enacting the definitions and rules embodied in said models, where there is an active model corresponding to each of said process models and data models;

runtime objects comprising objects containing the runtime state of instances of said process models and data models, where there may be at any time any number of runtime objects instantiated from each of said process models and data models by the corresponding said active model; and

a model loader comprising an object responsible for loading said models from their formal representations stored in a repository, converting said loaded models to corresponding said active models, and caching said active models *(**Williams**: figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 41; for all above).*

Claim 19

Williams and **Parr** disclosed the runtime engine of claim 17, wherein the runtime engine executes each of said models as a series of processing steps, wherein:

each of said processing steps is triggered by the receipt of an external input;

the runtime engine invokes at least one instance of at least one relevant process model to handle said received input, and executes sub-processes of said invoked processes as defined by the relevant said flow rules; and

a processing step ends when any further activities to be performed depend on the receipt of other external inputs *(**Williams**: figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 41; for all above).*

Claim 20

Williams and **Parr** disclosed the runtime engine of claim 17, wherein:

the runtime engine executes each of said models as a series of processing steps by invoking at least one instance of said process models;

the full state of each of said at least one process instances is made persistent at the end of each said processing step;

execution of each of said at least one process instance can resume from its stored state at any relevant time; and

a repository of all said at least one process instances is available for queries and retrieval by the runtime engine while executing said models or by external

applications *(**Williams**: figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 41; for all above).*

Claim 24

Williams and **Parr** disclosed a method for substantially overcoming the need to write computer source code in order to develop software applications, comprising:

creating models of the applications in a visualizable computer executable modeling language system, using a visual modeling tool, comprising:

defining each of said software applications as a hierarchy of process models, slots, data models, and flow rules;

classifying some of said process models and said data models as atomic;

classifying some all other process models and data models as composite;

defining each of said composite process models as a construction of at least one of sub process models, slots, data models and flow rules;

defining each of said composite data models as a construction of at least one sub data model; and

defining each of said flow rules as connecting a pair of said slots, data models and sub data models, wherein said flow rules define both data flow and process flow; and executing the logic defined by said created models.

(as disclosed above under claims 1, 2 and 17).

Claim 25

Williams and **Parr** disclosed the method of claim 24, wherein the execution of the logic defined by said models is made by a dedicated computer program (runtime engine) *(as disclosed above under claims 1, 2 and 17).*

Claim 27

Williams and **Parr** disclosed a software development platform for substantially overcoming the need to write computer source code in order to develop software applications, comprising:

a visualizable computer executable modeling language of the definition of software solutions, said definition comprising:

defining each of said software solutions as a hierarchy of process models, slots, data models, and flow rules;

classifying some of said process models and said data models as atomic;

classifying all other process models and data models as composite;

defining each of said composite process models as a construction of at least one sub process modes, slots, data models and flow rules;

defining each of said composite data models as a construction of at least one sub data model; and

defining each of said flow rules as connecting a pair of said slots, data models and sub data models, wherein said flow rules define both data flow and process flow;

a visual modeling tool for defining said software solutions by at least one user as said hierarchies of models in said modeling language; and

a dedicated computer program to automatically execute said software solutions according to the logic defined by said hierarchies of models.

(as disclosed above under claims 1, 2 and 17)

Claim 28

Williams and **Parr** disclosed the software development platform of claim 27, wherein said dedicated computer program is a runtime engine that automatically executes said software solutions at runtime, according to the logic defined by said hierarchies of models.

(as disclosed above under claims 1, 2 and 17)

11. Claims 21-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Williams** (USPN 5,850,548) in view of **Goodwin** et al. (USPN 6,199,195).

Claim 21

Williams disclosed each of the applications is defined by a single said process model and a hierarchy of its sub-models *(figures 3A-4G; column 2, lines 20-39; and column 5, line 31 to column 13, line 42).* **Williams** did not explicitly state the code generator produces code in a general purpose programming language implementing the application exactly as defined by said single process model

and said hierarchy of its sub-models, thus substantially eliminating the need for writing code in any programming language to implement the application. **Goodwin** demonstrated that it was known at the time of invention to use models to generate code (figures 2 and 3). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the visual modeling system of **Williams** with code generation as found in **Goodwin**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to allow for rapid development and integration of components and services (**Goodwin**: column 8, lines 20-31).

Claim 22

Williams and **Goodwin** disclosed the software program of claim 21, wherein said general purpose programming language is Java *(**Goodwin**: figure 4).*

Claim 23

Williams and **Goodwin** disclosed the software program of claim 21, wherein said general purpose programming language is C++ *(**Goodwin**: column 12, lines 56).*

12. Claims 26 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Williams** (USPN 5,850,548) in view of **Parr** et al. (US 2002/0095653) in further view of **Goodwin** et al. (USPN 6,199,195).

Claim 26

Williams, **Parr** and **Goodwin** disclosed the method of claim 24, wherein the implementation of the logic defined by said models is made by the code of a software program in a general purpose programming language, which is generated by dedicated computer program (code generator) *(as above for claim 21, in view of **Williams** and **Parr** in claim 24).*

Claim 29

Williams, **Parr** and **Goodwin** disclosed the software development platform of claim 27, wherein the dedicated computer program is a code generator that automatically generates the code of a software program in a general purpose programming language implementing the logic defined by said hierarchies of models *(as above for claim 21, in view of **Williams** and **Parr** in claim 27).*

Response to Arguments

Applicant's arguments filed 23 August 2007 have been fully considered but they are not persuasive. Applicant argues the cited art fails to disclose: 1) data models as sub-components; 2) input slots as mandatory or optional; 3) mapping between database tables and data models; 4) various composition methods of composite data models; 5) code generation as in claim 17; and 6)

totally eliminating the need for writing code in any programming language. Applicant further argues cited art combination rely upon hindsight.

With regard to arguments one and four, **Williams** clearly disclosed composition of data models to make composite data models (column 3, lines 66). A composite message is at very least a collection. Further note column 5, lines 41-45.

With regard to argument two, **Williams** demonstrated optional and mandatory ports/slots as the above rejections indicate (figures 3A-B and 18; column 8, lines 45-46; column 17, lines 49-50; thus ports/slots are optional, when not connected and mandatory when connected and responding in a mandatory fashion to external events).

With regard to argument three, data models are linked/mapped to the database (storage of such data) through various names and relations among the objects (see at least: column 14, line 46-53 and column 21, line 59).

With regard to argument five, as **Parr** indicates visual representations being converted to execution and both **Parr** and **Williams** indicate limiting programming to visual programming, it is clear that the combination demonstrate generating code from the visual representations. The "complex" notions of the visual language are disclosed by **Williams** as indicated. **Parr** demonstrates a visual language being converted to execution.

With regard to argument six, both **Williams** and Applicant's originally filed disclosure seem to allude to some writing of source code. Further, both

the cited art and the original disclosure program/produce "code" in a visual programming language. Further, the limitation in question does not have support in the originally filed disclosure as previously indicated in the above rejections under 35 USC 112 and 102/103.

Finally, Applicant suggests motivation to combine the references is not adequate and thus based upon hindsight. This is not true as motivation is clearly cited from the cited prior art. Having addressed the issues raised by Applicant, the rejections are maintained as indicated.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

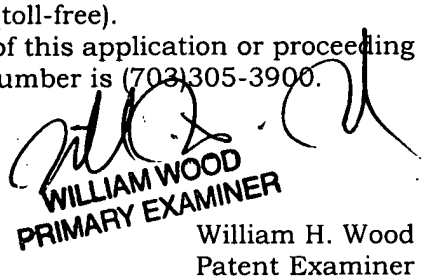
Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (571)-272-3736. The examiner can normally be reached 10:00am - 4:00pm Monday thru Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)-272-3756. The fax phone numbers for the organization where this application or proceeding is assigned are (571)273-8300 for regular communications.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained form either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR systems, see http://pair-direct.uspto.gov. For questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.



William H. Wood
Patent Examiner
AU 2193
November 9, 2007